November 2003

Decentralized Algorithms for Multirobot Manipulation via Caging

Guilherme A.S. Pereira
University of Pennsylvania

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Mario F.M. Campos
Universidade Federal de Minas Gerais

Postprint version. Published in *Springer Tracts in Advanced Robotics*, Volume 7, November 2003, pages 257-274. The original publication is available at www.springerlink.com.
Publisher URL: http://dx.doi.org/10.1007/b80173

Decentralized Algorithms for Multirobot Manipulation via Caging

Abstract
This paper addresses the problem of transporting objects with multiple mobile robots using the concept of *object closure*. In contrast to other manipulation techniques that are typically derived from form or force closure constraints, object closure requires the less stringent condition that the object be trapped or caged by the robots. Our basic goal in this paper is to develop decentralized control policies for a group of robots to achieve a condition of object closure, and then, move toward a goal position while maintaining this condition. We present experimental results that show car-like robots controlled using visual feedback, transporting an object in an obstacle free environment toward a prescribed goal.

Comments
Postprint version. Published in *Springer Tracts in Advanced Robotics*, Volume 7, November 2003, pages 257-274. The original publication is available at www.springerlink.com.
Publisher URL: http://dx.doi.org/10.1007/b80173

Decentralized Algorithms for Multirobot Manipulation via Caging

Guilherme A. S. Pereira[1,2], Vijay Kumar[1], and Mario F. M. Campos[2]

[1] GRASP Lab., University of Pennsylvania, Philadelphia, PA, USA.
[2] VERLab, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil.

Abstract. This paper addresses the problem of transporting objects with multiple mobile robots using the concept of *object closure*. In contrast to other manipulation techniques that are typically derived from form or force closure constraints, object closure requires the less stringent condition that the object be trapped or caged by the robots. Our basic goal in this paper is to develop decentralized control policies for a group of robots to achieve a condition of object closure, and then, move toward a goal position while maintaining this condition. We present experimental results that show car-like robots controlled using visual feedback, transporting an object in an obstacle free environment toward a prescribed goal.

1 Introduction

Object manipulation with mobile robots have been extensively discussed in the literature. Most approaches use the notions of force and form closure to perform the manipulation of relative large objects [1–4]. *Force closure* is a condition that implies that the grasp can resist any external force applied to the object [5]. *Form closure* can be viewed as the condition guaranteeing force closure, without requiring the contacts to be frictional. In general, robots are the agents that induce contacts with the object, and are the only source of grasp forces. But, when external forces acting on the object, such as gravity and friction, are used together with contact forces to produce force closure, we have a situation of *conditional force closure*. Several research groups have used conditional closure to transport an object by pushing it from an initial position to a goal [6,7].

In contrast to these approaches, as shown in Fig. 1, object closure requires the less stringent condition that the object be trapped or caged by the robots. (Our use of the concept of caging is slightly different from the definition in [8], and hence the object closure.) In other words, although the object may have some freedom to move, it cannot be completely removed [9,10]. Because a caging operation requires a relatively low degree of precision in relative positions and

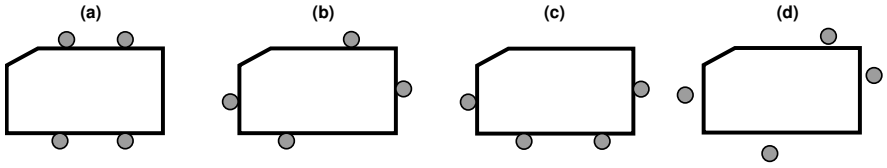

Fig. 1. Four technics of manipulation: (a) force closure (robots pressing the object); (b) form closure; (c) conditional closure (robots pushing the object up); (d) object closure or caging.

orientations, manipulation strategies based on caging operations are potentially more robust than, for example, approaches relying on force closure.

Caging was first introduced by Rimon and Blake in [8] for non-convex objects and two fingered gripers. Other papers addressing variation on this basic theme are [9–13]. Broadly speaking, our work may be considered closest to the work by Sudsang and Ponce [13]. They develop a centralized algorithm for moving three robots with circular geometry in an object manipulation task.

Our basic goal in this paper is to develop decentralized control policies for a group of robots to achieve a condition of object closure, and then, move toward a goal position while maintaining the object closure condition. Unlike previous work [13,10], we do not require the robots to be circular. However, we do introduce a number of simplifying assumptions to enable real-time implementation. Further, our interest is in transporting the object from an initial position toward a goal position in $\mathbb{R}^2$. We do not address the problem of precisely positioning and orienting the object in the plane.

The remaining of this paper is organized as follows. We first discuss the mathematical modeling of the object and the robots. Section 3 presents our definition of object closure, and several key necessary conditions for establishing and maintaining object closure. Section 4 describes our approach to cooperative control. We briefly present results from experiments in Section 6. Finally, the main points of the paper and directions for future work are presented in Section 7.

2 Mathematical Modeling

Consider a planar world, $\mathcal{W} = \mathbb{R}^2$, occupied by a convex, polygonal object $\mathcal{O}$, and a group of n convex, polygonal robots. The i^{th} robot R_i is described by the convex set $\mathcal{A}_i(q_i) \in \mathcal{W}$, where $q_i = (x_i, y_i, \theta_i)$

denotes the configuration of R_i. The configuration of the object is described by the coordinates $q = (x, y, \theta)$. We will use $\mathcal{C}_{\mathcal{R}_i}$ to denote the configuration space of a robot, while $\mathcal{C}$ will denote the configuration space for the object $\mathcal{O}$.

Convex robots and objects are represented by an intersection of m half planes derived from the equations for each edge. The edge from (x_j, y_j) to (x_{j+1}, y_{j+1})[1] is given by $f_j(x, y) = a_j x + b_j y + c_j$ and $f_j(x, y) < 0$ for all points in the interior of the polygon.

If robot positions and orientations are held fixed, the region in the configuration space that corresponds to an interpenetration between the object $\mathcal{O}$ and the robot i is:

$$\mathcal{C}_{obj\_i} = \{q \in \mathcal{C} \mid interior(\mathcal{A}_i(q_i) \cap \mathcal{O}(q) \neq \emptyset)\}, \tag{1}$$

where $\mathcal{O}(q)$ is the representation of $\mathcal{O}$ in the configuration q.

3 Object Closure

3.1 Definition

Before we proceed further, we will make three assumptions in this section.

Assumption A1. All robots are holonomic and identical in terms of geometry, and in terms of capabilities and constraints related to sensing, control, and mobility.

Assumption A2. All robots are point robots – $\mathcal{A}_i(q_i) = q_i = (x_i, y_i)$.

Assumption A3. The manipulated object cannot rotate – the coordinates of the object are given by $q = (x, y)$ and $\mathcal{C} \subset \mathbb{R}^2$.

Assumptions *A2* and *A3* make it easier to explain the basic ideas and will be relaxed in the next sections.

Figure 2 shows the boundary of $\mathcal{C}_{obj\_i}$ for a five-sided polygonal object and the point robot, R_i.

The union of $\mathcal{C}_{obj\_i}$ for $1 \leq i \leq n$ determines the region in $\mathcal{C}$ which cannot be occupied by the object. Then,

$$\mathcal{C}_{obj} = \bigcup_{i=1}^{n} \mathcal{C}_{obj\_i}. \tag{2}$$

[1] $j + 1$ is replaced by 1 for $j = m$ and $j - 1$ by m for $j = 1$.

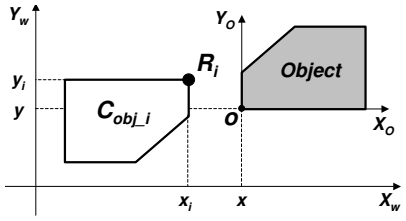

Fig. 2. $\mathcal{C}_{obj\_i}$ for a point robot R_i considering only object translations. By sliding the object around the robot, the origin, o, of the object-fixed reference frame traces out the boundaries of $\mathcal{C}_{obj\_i}$.

Let the complement of $\mathcal{C}_{obj}$ in $\mathcal{C}$ be $\bar{\mathcal{C}}_{obj}$. When $\bar{\mathcal{C}}_{obj}$ consists of two (or more) disjoint sets, we use the term *object closure* to refer to the condition when one of these sets is compact and contains the object configuration, q. This is shown for four robots in Fig. 3, where the compact set, which we refer to as the *closure configuration space* and denote by $\mathcal{C}_{cls}$, is shown shaded. Observe that the object is trapped or caged (in the terminology of [8]) when its origin is in $\mathcal{C}_{cls}$.

We can easily relax Assumption *A3* to accommodate the more general case with translations and rotations. In this case, (2) remains the same, but $\mathcal{C}_{obj\_i}$ in (1) is a three-dimensional solid whose cross-section, for a given angular orientation, is similar to the picture in Fig. 2, and the compact subset $\mathcal{C}_{cls}$ consists of one or more three dimensional solids whose cross-section (slice) is similar to the one shown in Fig. 3 [10].

We now define a *non essential robot* with the help of Fig. 4. In contrast to Fig. 3 in which all four $\mathcal{C}_{obj\_i}$ (and therefore all four robots) are essential to construct the boundary for the closure configuration space, R_3 is not essential for object closure in Fig. 4. In a group of robots maintaining object closure, a non essential robot, R_x, is a robot whose removal (and consequently the absence of the constraint due to $\mathcal{C}_{obj\_x}$) does not violate the state of object closure.

We now introduce a fourth assumption:

Assumption A4. We will only consider groups of robots with no non essential robots.

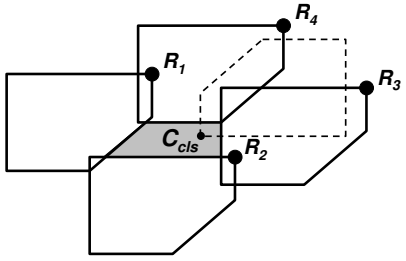

Fig. 3. Object closure: The interior (shaded gray) represents the *closure configuration space*, $\mathcal{C}_{cls}$, for a team of 4 robots. The dashed polygon represents the object. Notice that the origin of the object's reference frame is inside $\mathcal{C}_{cls}$, a compact set, indicating a condition of object closure.

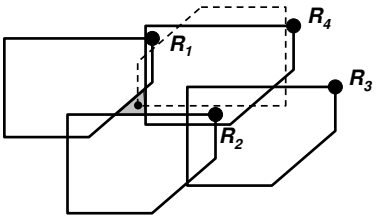

Fig. 4. Essential Robots: Even with the removal of R_3 the closure properties of the group is preserved and so, R_3 is a non essential robot.

3.2 A Test for Object Closure

Checking the object closure condition involves two steps: (a) Establishing the existence of $\mathcal{C}_{cls}$; and (b) Verifying $q \in \mathcal{C}_{cls}$. Step (a) requires obtaining state information from all robots and Step (b) requires obtaining position (pose, in the more general case) of the object.

The key idea comes from Figure 3 where robots are numbered R_1 through R_n in a counter clockwise fashion. A necessary condition for object closure with non essential robots is that the i^{th} robot's position satisfies: $\mathcal{C}_{obj\_i-1} \cap \mathcal{C}_{obj\_i} \neq \emptyset$; and $\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_i+1} \neq \emptyset$. This condition is not sufficient. The sufficient condition involves verifying $q \in \mathcal{C}_{cls}$. However, this condition is necessary and sufficient for the existence of $\mathcal{C}_{cls}$ (step (a) above), and for maintaining object closure. Hence we can state the following:

Proposition 1. *If an object is in a state of object closure with a group of robots with no non essential robots, a sufficient condition for maintaining object closure is* $\mathcal{C}_{obj\_i-1} \cap \mathcal{C}_{obj\_i} \neq \emptyset$*; and* $\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_i+1} \neq \emptyset$.

We now explain how to derive the algebraic equations for object closure. Define $\mathcal{I}_i$ to be a set of the robots configuration space that represents the intersection between $\mathcal{C}_{obj\_i}$ and $\mathcal{C}_{obj\_k}$:

$$\mathcal{I}_i = \{q_k \in \mathcal{C}_{R_k} \mid \mathcal{C}_{obj\_i}(q_i) \cap \mathcal{C}_{obj\_k}(q_k) \neq \emptyset\},$$

Note that $\mathcal{C}_{obj\_i}(q_i)$ and $\mathcal{C}_{obj\_k}(q_k)$ are identical polygons, which introduces a symmetry in the form of $\mathcal{I}_i$.

It can be observed that:

$$\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_k} \neq \emptyset \Leftrightarrow (q_k \in \mathcal{I}_i \wedge q_i \in \mathcal{I}_k).$$

Thus, the object closure conditions for each robot, which can be rewritten as $q_i \in \mathcal{I}_{i-1}$ and $q_i \in \mathcal{I}_{i+1}$ (see Fig. 5(a)) are represented as a set of inequality constraints of the form $g_j(q_{i-1}, q_i) \leq 0$ or $g_j(q_i, q_{i+1}) \leq 0$,

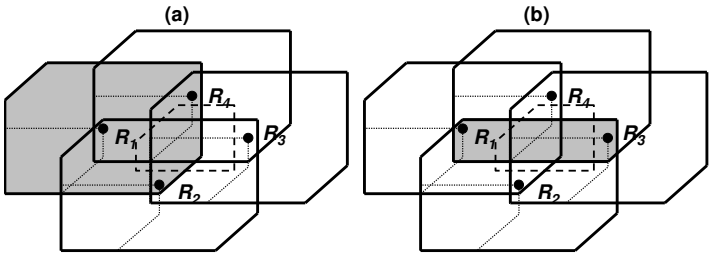

Fig. 5. Object closure is achieved if each robot i is inside Γ_i. The shaded areas represent (a) $\mathcal{I}_1$ and (b) Γ_1.

where g_j are the functions that delimit $\mathcal{I}_{i-1}$ or $\mathcal{I}_{i+1}$ respectively. $\mathcal{I}_{i-1}$ ($\mathcal{I}_{i+1}$) is a $2m$-sided polygon defined by $2m$ algebraic constraints , each linear in q_{i-1} and q_i (q_i and q_{i+1}). Since each polygon has up to $2m$ sides, the number of constraints for each robot is $4m$. For the situation we are considering, where the robots are points and the object cannot rotate, the boundary of $\mathcal{C}_{obj\_i}$ is formed by the same edges of $\mathcal{O}$ but ordered in a different way (see Fig. 2). Then, each $\mathcal{I}_i$, which depends on $\mathcal{C}_{obj\_i-1}$ and $\mathcal{C}_{obj\_i+1}$, is bounded by two sets of the object's edges (refer to the algorithm presented in [14] for proofs). Consequently, $\mathcal{I}_{i-1}$ is given by functions $g_j(q_{i-1}, q_i)$, while $\mathcal{I}_{i+1}$ is given by another set of functions $g_j(q_i, q_{i+1})$. Each function is directed derived from the functions $f_i(x, y)$ used to describe the object.

We can now rewrite Proposition 1 as follows:

Proposition 2. *If an object is in a state of object closure with a group of robots with no non essential robots, a sufficient condition for maintaining object closure is $q_i \in \Gamma_i$, where $\Gamma_i = \mathcal{I}_{i-1} \cap \mathcal{I}_{i+1}$.*

Note Γ_i is bounded by a subset of the constraints $g_j(q_i) \leq 0$, $1 \leq j \leq 4m$. A example of Γ_i can be seen in Fig. 5(b).

3.3 Introducing Rotations

Thus far, we have ignored rotations. In reality, since the robots will collide with and bump against the object, the object can rotate. Even if object closure is guaranteed for a given object orientation, a small rotation followed by a translation may cause the object to escape from the robot formation.

Our approach to incorporate rotations is to establish guarantees for object closure under the worst case rotation. Because the object

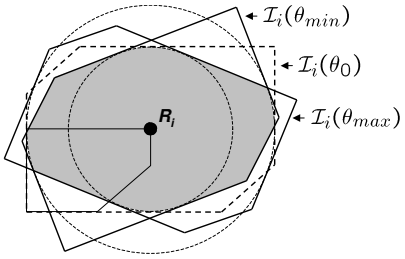

Fig. 6. Closure region for a maximum rotation of 20°.

has no actuators, its maximum velocity is limited by the maximum velocity of the robots. Thus, if the object orientation at any instant is estimated to be θ_o, the orientation in the ensuing interval ΔT must be in the interval, $[\theta_{min}, \theta_{max}]$, where $\theta_{min} = \theta_o - \Delta T\,\omega_{max}$, $\theta_{max} = \theta_o + \Delta T\,\omega_{max}$, and ω_{max} is the (estimated) maximum object's angular velocity. Let $\mathcal{J}_i$ be defined as:

$$\mathcal{J}_i = \bigcap_{\theta=\theta_{min}}^{\theta_{max}} \mathcal{I}_i(\theta)\,,$$

where $\mathcal{I}_i(\theta)$ is $\mathcal{I}_i$ computed for an object orientation θ. Following the previous methodology, the conditions that guarantee object closure for all $\theta \in [\theta_{min}, \theta_{max}]$ are: $q_i \in \mathcal{J}_{i-1}$ and $q_i \in \mathcal{J}_{i+1}$.

Since $\mathcal{C}_{obj\_i}$ is represented by the same polygon for every robot, the shape of $\mathcal{I}_i(\theta)$ is independent of the object orientation. As θ changes, $\mathcal{I}_i(\theta)$ is obtained by simply rotating $\mathcal{I}_i(\theta_o)$ around R_i. The intersection set $\mathcal{J}_i$, can be constructed as shown in Fig. 6. The shaded area represents the configuration space where q_i must be in order to guarantee object closure for object orientations between θ_{min} and θ_{max}. It is bounded by circular arcs and the sides of $\mathcal{I}_i(\theta_{min})$ and $\mathcal{I}_i(\theta_{max})$. Notice that we continue with a set of inequalities constraints but now two of them are nonlinear. The set $\mathcal{J}_i$ is still convex. From a practical standpoint, this set-valued approach for modeling the uncertainty in orientation allows us to be robust to errors in pose estimation.

3.4 Working with Polygonal Robots

The main difficulty of working with polygonal robots is the computation of $\mathcal{C}_{obj\_i}$ in real time. Although an efficient algorithm exists, it must run every time step since changes in robot orientation alters not only the orientation but also the shape of $\mathcal{C}_{obj\_i}$. Furthermore, differences on the shapes of $\mathcal{C}_{obj\_i}$ and its neighbors make the form of closure

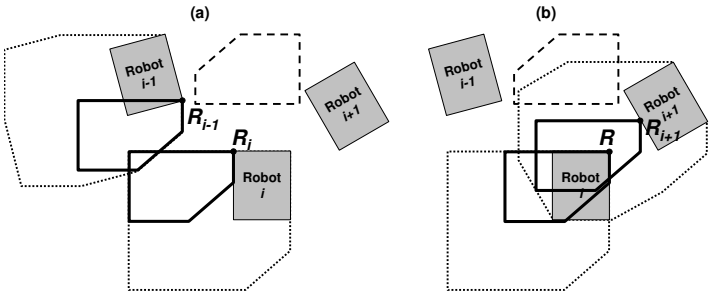

Fig. 7. Robot i checks closure (a) using the imaginary point robots, R_i and R_{i-1} (left) (b) using a different set of point robots, R_i and R_{i+1} (right). The dotted polygons are the actual object configuration space.

constraints very complicated. However, since $\mathcal{C}_{obj\_i}$ can be constructed by the union of the $\mathcal{C}_{obj\_i}$ of the points in the convex polygon that form the robot, we can easily establish a sufficient condition that guarantees closure. If the intersection between $\mathcal{C}_{obj\_i}$ of two virtual point robots located at the closest pair of points between robot k and robot $k+1$ is non zero, $\mathcal{C}_{obj\_k} \cap \mathcal{C}_{obj\_k+1} \neq \emptyset$. Since $\mathcal{C}_{obj\_i}$ of a point robot can be computed off-line, the online computation is limited to the translation of this set to the location of the virtual point robots. This computation is illustrated in Fig 7.

For each slice of the configuration space (a specific orientation), the object configuration space for the polygonal robot k can be written as:

$$\mathcal{C}_{OBJ\_k} = \mathcal{A}_k(q_k) \oplus \mathcal{C}_{obj\_i}(0) \ ,$$

where $\mathcal{C}_{obj\_i}(0)$ is the the object configuration space for the point robot located in the origin of the world reference frame and $\oplus$ is the Minkowski sum operator. We can observe also that:

$$\mathcal{C}_{obj\_i}(q) = \{q\} \oplus \mathcal{C}_{obj\_i}(0) \ .$$

Thus, it is not too hard to verify:

Proposition 3. *If $\mathcal{C}_{obj\_i}(q_a) \cap \mathcal{C}_{obj\_i}(q_b) \neq \emptyset$, and $q_a \in \mathcal{A}_k(q_k)$ and $q_b \in \mathcal{A}_l(q_l)$ are the closest pair of points between the two convex robots, then:*

$$(\mathcal{A}_k(q_k) \oplus \mathcal{C}_{obj\_i}(0)) \cap (\mathcal{A}_l(q_l) \oplus \mathcal{C}_{obj\_i}(0)) \neq \emptyset \, .$$

Thus, using the closest pair of points as reference for our computations leads us to a conservative but simple test for object closure for polygonal robots.

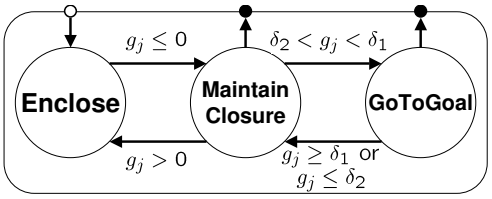

Fig. 8. The switched control system with 3 modes for multirobot manipulation. δ_1 and δ_2 are thresholds for activating the transition between modes.

4 Control

Our control system is decentralized and implemented using a set of reactive controllers. Each robot switches between the controllers as shown in Fig. 8. The switches are governed by the activation of constraints that depend on the relative positioning of a robot with respect to its neighbors and the robots' estimate of the object orientation.

Recall from Proposition 2, that object closure constraints for R_i are defined by inequalities, $g_j(q_{i-1}, q_i) \leq 0$ or $g_j(q_i, q_{i+1}) \leq 0$. We consider the j^{th} constraint to be active when $g_j = \delta_1$, where δ_1 is a small negative number that can be thought of as a threshold. In addition to ensuring $q_i \in \Gamma_i$, it is also necessary to ensure the robots do not try to cluster together thus crushing the object. From a practical standpoint, although the object may be rigid and immune to damage, this will cause large contact forces and jamming due to friction. To avoid this, we introduce a new set of constraints that prevent the robot from being very close of its neighbors: $g_j \geq \delta_2$, where $\delta_2 < \delta_1 < 0$. This defines a "safe" configuration space for each q_i where the object is caged but jamming is avoided. Practically, the set of constraints $\delta_2 \leq g_j \leq \delta_1$ define two polygons with the same shape as Γ_i but with different sizes. The reactive controllers and the sequential composition of these controllers are shown in Fig. 8.

We consider a simple kinematic model for each robot. For the i^{th} robot, the dynamical model is given by:

$$\dot{q}_i = u_i \,,$$

where $q_i = (x_i, y_i)$.

We will denote the constraints due to R_{i-1}, which have the form $g_j(q_{i-1}, q_i) \leq 0$, by g_j^l and those due to R_{i+1}, which have the form $g_j(q_i, q_{i+1}) \leq 0$ by g_j^r. In the *Enclose* mode each robot tries to initially

achieve object closure. The control input in this mode is:

$$u_i = -k_1 \left[a \nabla g_j^r + b \nabla g_j^l \right] , \tag{3}$$

where ∇g_j^x is the gradient of the constraint defined as $\partial g_j^x / \partial q_i$. ∇g_j^l is due to robot $i-1$ and ∇g_j^r is due to robot $i+1$. We assume that these vectors are normalized and are of unit length. The variables a and b can each be -1, 0, or 1. When $g_j \leq \delta_2$ the value -1 is assigned. When $\delta_1 > g_j > \delta_2$, the value 0 is assigned. And when $g_j \geq \delta_1$, the value 1 is assigned.

In the *MaintainClosure* mode a robot tries to maintain object closure while navigating towards the goal. The control input for this state is:

$$u_i = -k_1 \left[a \nabla g_j^r + b \nabla g_j^l \right] + k_2 \, u_T \, , \tag{4}$$

where u_T is a common input to the team of robots that leads them to the goal. It can be, for example, a proportional control law that is a function of the object position, $q - q_{goal}$. Alternatively, u_T can be derived from a potential function that also serves as a Lyapunov function. We will assume that

$$u_T = -\nabla \phi(q) \, , \tag{5}$$

where $\phi(q)$ is a potential function with a single minimum at q_{goal}.

The *GoToGoal* mode has the following input:

$$u_i = k_2 \, u_T \, . \tag{6}$$

It is clear that we can always design u_T in (5) to guarantee that each robot will go to the specified target. Although, if one of the constraints $g_j \geq \delta_1$ or $g_j \leq \delta_2$ is violated, the controller switches to the *MaintainClosure* mode in (4). We will now prove that this controller guarantees that the condition of object closure is maintained.

Before we do that, we will need to make an important observation about the constraints $g_j(q_i, q_k)$. Let us consider, for illustration purposes, the special case of object translation. The set of constraints, Γ_i, for the point robot i can be subdivided in two sets of linear inequalities. The equations of each set are of the form $a_j(x_i + x_k) + b_j(y_i + y_k) + c_j \leq 0$, where $k = i - 1$ (robot on the left) for the first set and $k = i + 1$ (robot on the right) for the second set. Observe that each constraint

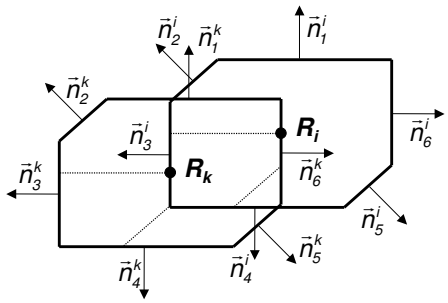

Fig. 9. An active constraint for robot i ($g_6^k(q_i, q_k) = 0$) indicates the activation of a identical constraint with opposite sign for one of its neighbors ($g_3^i(q_i, q_k) = 0$). In this picture, $\delta_1 = 0$. Notice that the normal vector of the active constraint for robot i, $\boldsymbol{n}_3^i$, is equal to $-\boldsymbol{n}_6^k$, the negative of the normal vector of the active constraint for robot k.

describes a line in the world reference frame translated by the position of one of the neighbors. Since $\mathcal{I}_i$ has the same form for all robots, when one constraint, $g_j(q_i, q_k)$, is active for one robot, there is an identical constraint with opposite sign, $-g_j(q_i, q_k)$, active for one of its neighbors. Figure 9 shows a typical situation when one constraint is active for the i^{th} robot and a identical constraint, with opposite sign, is active for one of its neighbors. This is also the case when $g_j(q_i, q_k)$ is not a linear function (as is the case when rotations are considered). In other words,

$$\frac{\partial g_j}{\partial q_k} = -\frac{\partial g_j}{\partial q_i} \tag{7}$$

We use this observation to prove that, once the robots have captured the object, the controller (4) guarantees object closure is maintained.

Proposition 4. *Once the robots achieve a situation of object closure the switched control system shown in Fig. 8 guarantees object closure.*

Proof. We need to prove that every time a generic constraint, $g_j(q_i, q_k) \leq \delta_1$, is active the control input makes $\dot{g}_j(q_i, q_k) \leq 0$. The time derivative of $g_j(q_i, q_k)$ is given by:

$$\dot{g}_j(q_i, q_k) = \frac{\partial g_j}{\partial q_i}\dot{q}_i + \frac{\partial g_j}{\partial q_k}\dot{q}_k \tag{8}$$

Denote $\partial g_j/\partial q_i$ by ∇g_j^k (induced by R_k), and $\partial g_j/\partial q_k$ by ∇g_j^i. Because of (7), $\nabla g_j^i = -\nabla g_j^k$. Assuming that for the i^{th} robot $g_j(q_i, q_k)$, whose gradient is ∇g_j^k, is active, then for the k^{th} robot, $g_p(q_i, q_k)$, whose gradient is $\nabla g_p^i = -\nabla g_j^k$, is also active. If, for example, R_k is the left neighbor of R_i then, in (4), $\nabla g_j^l = \nabla g_j^k$ for R_i and $\nabla g_j^r = \nabla g_p^i = -\nabla g_j^k$

for R_k. Substituting $\dot{q}_i$ and $\dot{q}_k$ in (8) by the control inputs in (4) (call ∇g_α the constraint induced by the other neighbor of R_i and ∇g_β the constraint induced by the other neighbor of R_k) we rewrite the time derivative of $g_j(q_i, q_k)$ as:

$$\begin{aligned}
\dot{g}_j(q_i, q_k) = &- k_1 \left[\nabla g_j^k \nabla g_j^k + a \nabla g_j^k \nabla g_\alpha\right] + k_2 \nabla g_j^k u_T + \\
&- k_1 \left[(-\nabla g_j^k)(-\nabla g_j^k) + b(-\nabla g_j^k)\nabla g_\beta\right] + \\
&+ k_2 \,(-\nabla g_j^k) u_T \\
= &- k_1 \left[2\nabla g_j^k \nabla g_j^k + a \nabla g_j^k \nabla g_\alpha - b \nabla g_j^k \nabla g_\beta\right] \\
= &- k_1 \left[2\|\nabla g_j^k\|^2 + a\|\nabla g_j^k\| \, \|\nabla g_\alpha\| \cos\theta_a + \right. \\
&\left. -b\|\nabla g_j^k\| \, \|\nabla g_\beta\| \cos\theta_b\right] \\
= &- k_1 \left[2 + a\cos\theta_a - b\cos\theta_b\right] \leq 0 \,.
\end{aligned}$$

Since $-1 \leq a\cos\theta \leq 1$, for all $a \in \{-1, 0, 1\}$, $\dot{g}_j(q_i, q_k) \leq 0$. Therefore, given the initial conditions, $g_j(q_i, q_k) \leq 0$, for all $1 \leq i \leq n$, $k \in \{q-1, q+1\}$ and $1 \leq j \leq 4m$. $\square$

Using the same methodology presented above we can also prove that when a constraint $g_j(q_i, q_k) = \delta_2$, the control law in the *MaintainClosure* mode maintains the condition $g_j(q_i, q_k) \geq 0$. Thus, if the robots are in the *MaintainClosure* mode, they either stay in this mode while moving toward the goal (the u_T component guarantees this [16]), or they switch to the *GoToGoal* mode.

It is more difficult to prove that the control law (3) in the *Enclose* mode leads to a condition of object closure. The main difficulties come from the assumption related to non essential robots and the book-keeping associated with numbering the robots. However, simple potential field controllers, like the one presented in [15] have the attractive property of symmetrically distributing the robots around the object and producing initial conditions that are favorable for the *Enclose* mode.

It is also natural to ask if the kinematic model can be extended to non-holonomic robots. For non-holonomic robots, u_i, which is a two dimensional vector, can be used as a set-point for controllers that take in account the non-holonomic constraints. An example of such an approach is shown in [16]. This is a direction of future research.

5 Computational Complexity

Considering that $\mathcal{I}_i$ (or $\mathcal{J}_i$ when rotations are considered) can be computed off-line, the decentralized manipulation algorithm consists in (a) determining the object orientation θ and the two neighbors' position; (b) computing $\mathcal{I}_i(\theta)$ by rotating $\mathcal{I}_i$; (c) translating $\mathcal{I}_i(\theta)$ to q_{i-1} and q_{i+1} and computing $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$ respectively; (d) verifying if there are active constraints; and (e) computing the control signals.

The estimation of θ, q_{i-1} and q_{i+1} is not addressed here but it is important to mention that in case of polygonal robots an $O(l)$ [17] algorithm needs to be used to determine the closest pair of points between the robots. Since $\mathcal{I}_i$ ($\mathcal{J}_i$) is defined by up to $2m$ functions, each robot needs to compute up to $2m$ rotations and $4m$ translations in order to compute $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$ ($\mathcal{J}_{i-1}$ and $\mathcal{J}_{i+1}$). The determination of the active constraints, if there are any, can be done by evaluating the $4m$ inequalities that define Γ_i. Observe however, that Γ_i don't need to be computed explicitly since the cost of computing this region is higher than evaluating all constraints for $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$. The computation of the control laws can be done in constant time. Therefore, the cost of the algorithm is $O(m+l)$, and is independent of the number of robots in the group.

6 Experiments

Our mobile robots are car-like robot equipped with an omnidirectional camera as its only sensor. Although we have performed experiments with teams of up to five mobile robots, we report here experiments with three robots. The communication among the robots relies on IEEE 802.11 networking. To facilitate the visual processing, each team member and the goal position are marked with different colors. Because each robot has only one camera we use communication between robots and cooperative sensing for (a) localization with respect to each other; and (b) estimating the pose of the object [18]. The communication is essentially used for multi-eyed stereo algorithms but is not used for control or decision making. Ground truth information is obtained from a calibrated overhead camera.

Figure 10 illustrates the test for object closure performed by Robot 1 (R_1). R_1 estimates the position of its neighbor R_2, as well as the orientation of the object. It then computes $\mathcal{C}_{obj\_1}$ and $\mathcal{C}_{obj\_2}$ based on its estimate of the pair of closest points, one on R_1 and one on R_2. As the

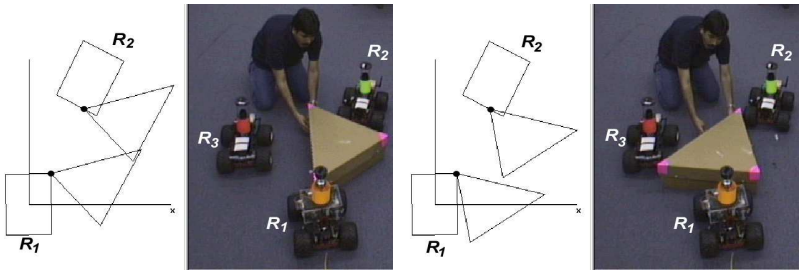

Fig. 10. Three robots caging a triangular object. R_1's computation of $\mathcal{C}_{obj\_1}$ and $\mathcal{C}_{obj\_2}$ for the imaginary point robots located at the closest pair of points are shown. The overlap (left) indicates the object is constrained for this specific orientation, and the lack of overlap (right) shows that object closure is not maintained for this slice of the configuration space.

figure shows, the snapshot on the left shows overlap and therefore a positive test for object closure. The snapshot on the right shows that the object can actually escape. A similar test (not shown in the figure) needs to be performed with robot R_3 .

Data collected from the overhead camera for typical experimental runs are shown in Figures 11 and 12. Robots R_1, R_2, and R_3 transport a triangular box toward a goal position. Figure 11 shows a situation where robots R_2 and R_3 change their control behaviors in order to perform the task. In Fig. 12, the actual $\mathcal{C}_{OBJ\_i}$ for the rectangular robot geometry is overlaid on the experimental data. Note, however, that the robots do not use the $\mathcal{C}_{OBJ\_i}$ for maintaining object closure, but instead they work with the virtual point robots as explained in Sect. 3.4. The object is caged in the three snap-shots shown.

7 Concluding Remarks

We presented algorithms for manipulating objects with multiple mobile robots combining the paradigms of pushing and caging. We defined the concept of object closure, a condition that ensures the objects are caged during manipulation. The main contributions of the paper are: (a) an algorithm that enable each robot to independently verify the condition of object closure; and (b) a decentralized control algorithm that enables each robot to move while maintaining object closure.

There are two main advantages of our approach. The decentralized algorithms only rely on the robots' ability to estimate the positions of their neighbors. Because robots are easily instrumented (in our case,

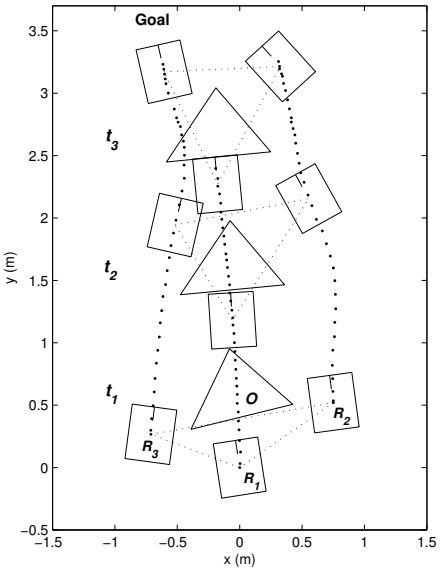

Fig. 11. Object transportation: t_1 – R_2 and R_3 are in the *Enclose* mode (see Fig. 8) trying to achieve object closure; t_2 – Object closure constraints are satisfied, R_2 and R_3 are in the *MaintainClosure* mode; t_3 – The robots are in the *GoToGoal* mode. R_1 is in the *GoToGoal* mode in all three snapshots.

this is done by tagging them with colored collars), this is relative easy even in an unstructured environment. Therefore, our methodology is potentially scalable for larger groups of robots operating in unstructured environments. Second, our algorithms do not rely on exact estimates of the position and orientation of the manipulated object. Therefore they are robust to errors in estimates of position and orientation of the manipulated object.

The main limitations of the algorithms used here include (a) the assumption of convex shapes; (b) the over approximation that is involved in verifying object closure when rotations are present; and (c) the use of the virtual point robots which result in sufficient conditions for maintaining object closure. All these result in conservative results with associated degradation in performance. For example, ensuring object closure with concave objects is often simpler than than is the case for convex objects. However, these assumptions and over approximations enable real-time performance and decentralized decision making with *guarantees*, and are important from a practical standpoint.

There are important directions of future work. First, it is necessary to explicitly model the nonholonomic behavior of the robots. The work in [16] provides a starting point in this direction. Second, we do not specifically consider algorithms for acquiring the object, establishing object closure (the *Enclose* mode) here. The papers [10,15]

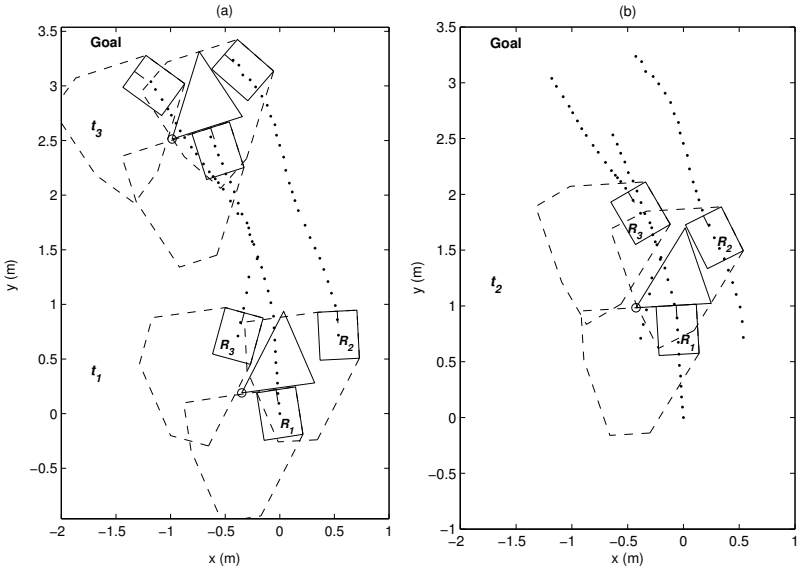

Fig. 12. The actual $\mathcal{C}_{OBJ\_i}$ (*dashed polygons*) for each robot. The origin of the object ($\circ$) is always inside $\mathcal{C}_{cls}$ (the compact set delimited by the three $\mathcal{C}_{obj\_i}$) indicating a object closure condition. (a) – initial and final configurations; (b) – an intermediate configuration.

provide some approaches to this, with guarantees for small (3-4) teams of robots. There are challenges in designing decentralized policies that scale up to large numbers of robots. One of the key steps here is to remove the assumption related to non essential robots. Finally, we do not address the precise positioning and orienting of the object. By varying the threshold δ_2, we can get tighter tolerances on the object position relative to the robots. However, it is also essential to plan trajectories for the individual robots, instead of simply prescribing a common feedforward control signal u_T. The work in [13] provides a starting point in this direction.

Aknowledments

We gratefully acknowledge the support of AFOSR grant no. F49620-01-1-0382, NSF grant no. CDS-97-03220, NSF grant no. IIS-0083420, DARPA ITO MARS Program, grant no. 130-1303-4-534328-xxxx-2000-0000, and CNPq-Brazil grants 200765/01-9 and 300212/99-2.

References

1. Ota J., Miyata N., Arai T. (1995) Transferring and regrasping a large object by cooperation of multiple mobile robots. In: *IEEE/RJS Int'l. Conf. on Intelligent Robots and Systems*, 543–548.
2. Kosuge K., Oosumi T. (1996) Decentralized control of multiple robots handling an object. In: *IEEE/RJS Int'l. Conf. on Intel. Robots and Systems*, 318–323.
3. Rus D. (1997) Coordinated manipulation of objects in a plane. *Algorithmica*, 19(1/2):129–147.
4. Sugar T., Kumar V. (1998) Decentralized control of cooperating mobile manipulators. In: *IEEE Int'l. Conf. on Robotics and Automation*, 2916–2921.
5. Rimon E., Burdick J. (1996) On force and form closure for multiple finger grasps. In: *IEEE Int'l. Conf. on Robotics and Automation*, 1795–1800.
6. Mataric M., Nilsson M., and Simsarian K. (1995) Cooperative multi-robot box-pushing. In: *IEEE/RJS Int'l. Conf. on Intelligent Robots and Systems*, 556–561.
7. Lynch K. M. (1996) Stable pushing: Mechanics, controllability, and planning. *Int'l. Journal of Robotics Research*, 15(6):533–556.
8. Rimon E., Blake A. (1996) Caging 2D bodies by one parameter, two-fingered gripping systems. In: *IEEE Int'l. Conf. on Robotics and Automation*, 1458–1464.
9. Davidson C., Blake A. (1998) Caging planar objects with a three-finger one-parameter gripper. In: *IEEE Int'l. Conf. on Robotics and Automation*, 2722–2727.
10. Wang Z., Kumar V. (2002) Object closure and manipulation by multiple cooperative mobile robots. In: *IEEE Int'l. Conf. on Robotics and Automation*, 394–399.
11. Sudsang A., Ponce J. (1998) On grasping and manipulating polygonal objects with disc-shaped robots in the plane. In: *IEEE Int'l. Conf. on Robotics and Automation*, 2740–2746.
12. Sudsang A., Ponce J., Hyman M., Kriegman D. J. (1999) On manipulating polygonal objects three 2-DOF robots in the plane. In: *IEEE Int'l. Conf. on Robotics and Automation*, 2227–2234.
13. Sudsang A., Ponce J. (2000) A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. In: *IEEE Int'l. Conf. on Robotics and Automation*, 1068–1075.
14. Latombe J-C. (1991) *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA.
15. Song P., Kumar V. (2002) A potential field based approach to multi-robot manipulation. In: *IEEE Int'l. Conf. on Robotics and Automation*, 1217–1222.
16. Esposito J. M., Kumar V. (2002) A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime. In: *IEEE Int'l. Conf. on Robotics and Automation*, 1691–1696.
17. Ponamgi M. K., Manocha D., Lin M. C. (1997) Incremental algorithms for collision detection between polygonal models. *IEEE Trans. on Visualization and Computer Graphics*, 3(1):51–64.
18. Pereira G. A. S., Kumar V., Spletzer J., Taylor C. J., Campos M. F. M. (2002) Cooperative transport of planar objects by multiple mobile robots using object closure. In: Siciliano B., Dario P. (Eds.) *Experimental Robotics VIII, 8th Int'l Symp. on Experimental Robotics*, Springer, 275–284.